Ejercicios de Python tipo coloquio

1. Leer de teclado (usando la función raw_input) los datos de un listado de alumnos terminados con padrón 0. Para cada alumno deben leer:

*) Padrón
*) Nombre
*) Apellido
*) Nota del primer parcial
*) Nota del primer recuperatorio (en caso de no haber aprobado el parcial)
*) Nota del segundo recuperatorio (en caso de no haber aprobado en el primero)
*) Nombre del grupo
*) Nota del TP 1
*) Nota del TP 2

Si el padrón es 0, no deben seguir pidiendo el resto de los campos.
Tanto el padrón, como el nombre y apellido deben leerse como strings (existen padrones que comienzan con una letra b), pero debe validarse que se haya ingresado algo de por lo menos 2 caracteres.
Todas las notas serán números enteros entre 0 y 10, aunque puede ser que el usuario accidentalmente ingrese algo que no sea un número, por lo que deberán validar la entrada y volver a pedirle los datos al usuario hasta que ingrese algo válido. También deben validar que las notas pertenezcan al rango de 0 a 10.
Se asume que todos los alumnos se presentan a todos los parciales hasta aprobar o completar sus
Al terminar deben:

a.1. imprimir por pantalla un listado de todos los alumnos en condiciones de rendir coloquio (último parcial aprobado y todos los TP aprobados) en el mismo orden en el que el usuario los ingreso.

a.2. imprimir por pantalla un listado de todos los alumnos en condiciones de rendir coloquio (último parcial aprobado y todos los TP aprobados) ordenados por padron en forma creciente.

a.3. imprimir por pantalla un listado de todos los alumnos en condiciones de rendir coloquio (último parcial aprobado y todos los TP aprobados) ordenados por nota y, en caso de igualdad, por padrón (ambos en forma creciente).

b. Calcular para cada alumno el promedio de sus notas del parcial y luego el promedio del curso como el promedio de todos los promedios.

c. Informar cuál es la nota que más se repite entre todos los parciales (sin importar si es primer, segundo o tercer parcial) e indicar la cantidad de ocurrencias.

d. listar todas las notas que se sacaron los alumnos en el primer parcial y los padrones de quienes se sacaron esas notas con el siguiente formato:
Nota: 2

* nnnn1
* nnnn2
* nnnn3
* nnnn4

Nota: 4
 * nnnn1
 * nnnn2
 ...
    Tener en cuenta que las notas pueden ser del 2 al 10 y puede ocurrir que nadie se haya sacado esa nota (y en dicho caso no esa nota no tiene que aparecer en el listado)
   e. suponiendo que se ingresa la información ordenada por grupo, indicar los grupos para los que todos los alumnos estén en condiciones de rendir coloquio sin leer más de una vez el vector de alumnos (más allá de la que se usó para cargar la información).

Hacer todos los procedimientos por separado.
Bajo ningún concepto el programa puede fallar.

2. Leer de un archivo CSV (de texto) en el que cada línea tendrá el siguiente formato:

padron,nombre,apellido,nota_parcial,nombre_de_grupo,nota_tp1,nota_tp2

Y resolver los mismos puntos que antes, pero contemplando que los alumnos ahora sólo tendrán una nota de parcial (la del último que hayan rendido).
Puede ocurrir que algunos registros no tengan con todos los campos, o que los campos numéricos no sean números, o que no pertenezcan al rango de 0 a 10. En dichos casos se deberán guardar esas líneas para mostrarlas una vez leído todo el archivo indicando que tienen algún error (no es necesario especificar cuál es el error).
Si bien el nombre del archivo lo ingresa el usuario, se puede asumir que el archivo existe.

3. Suponiendo que existe un archivo llamado utils.py donde se encuentran las funciones:

```
def guardar_en_archivo(archivo, contenido):
    """Guarda lo que le pasen como segundo parámetro en el archivo que
    recibe como primer parámetro.
    archivo tiene que estar abierto en modo binario y para escritura (wb)
    """
    ...


def leer_desde_archivo(archivo):
    """Lee del archivo archivo un registro y lo retorna junto con una
    variable booleana que indica si llegó al fin de archivo o no.
    archivo tiene que estar abierto en modo binario y para lectura (rb)
    """
    ...
    return data, fin_de_archivo
```

Leer dos archivos (61_matematica.dat y 75_computacion.dat) que tendrán registros con los campos:

* padron
* nombre
* apellido
* nota
* codigo_departamento
* codigo_materia

y armar uno nuevo donde sólo figuren las notas de los alumnos aprobados ordenados por padrón.
Ambos archivos están ordenados por padrón y se deben leer una única vez. Como los archivos pueden ser muy grandes, no se pueden guardar en memoria.
Una vez procesados los dos archivos se tienen que informar, para cada materia, cuántos alumnos aprobaron y cuántos desaprobaron.